MAGIS Atom Simulation Updates

Northwestern/SLAC

Sean Gasiorowski* (SLAC)

On behalf of the SLAC MAGIS team (Michael Kagan*, Ariel Schwartzman, Sanha Cheong, Murtaza Safdari, Maxime Vandegar, Omer Rochman) and the Northwestern MAGIS team (Natasha Sachdeva*, Yiping Wang*, Tim Kovachy, Zilin Chen, Jonah Glick, Arthur Pierce)

Science and Simulation Meeting

August 3rd, 2022

* active contributor to today's results

SLAC NATIONAL ACCELERATOR LABORATORY

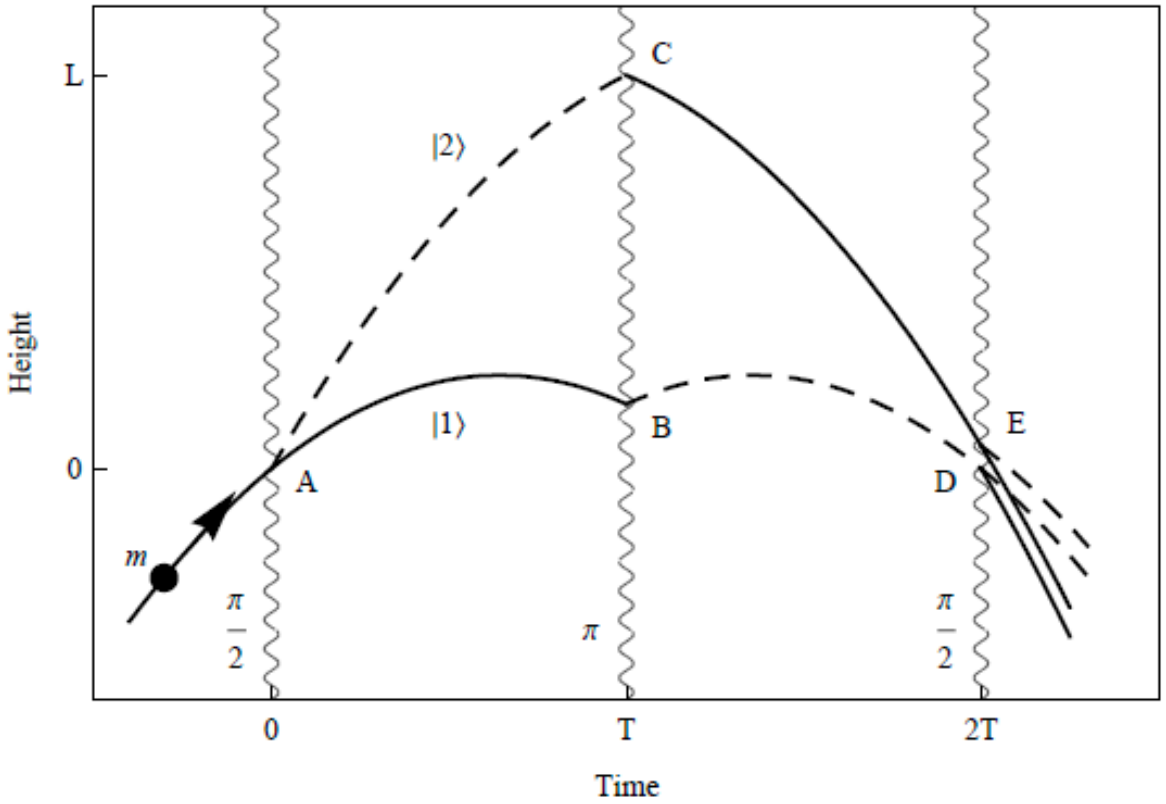
Stanford University | U.S. DEPARTMENT OF ENERGY

Introduction and Problem Setup

Physics framework: **semi-classical approximation**

- Per-atom simulation of trajectories in atom interferometer
  - Trajectories are solved using classical equations of motion
  - Phase difference between interferometer arms is calculated using information from classical trajectories
    - Phase probabilities used for probabilistic assignment to two ports
- Parameters used in simulation are for **point source interferometry**

Simulator written (by Natasha and the Northwestern group) in Julia

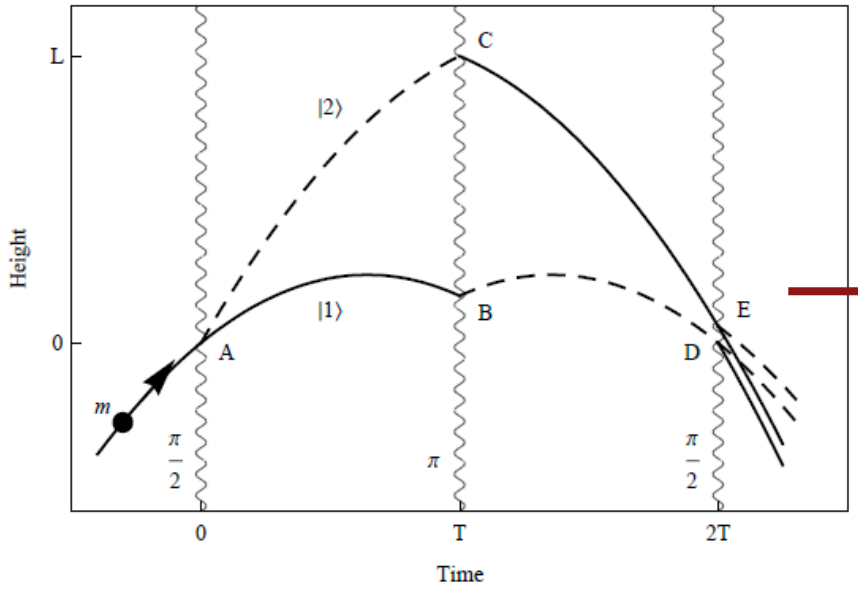


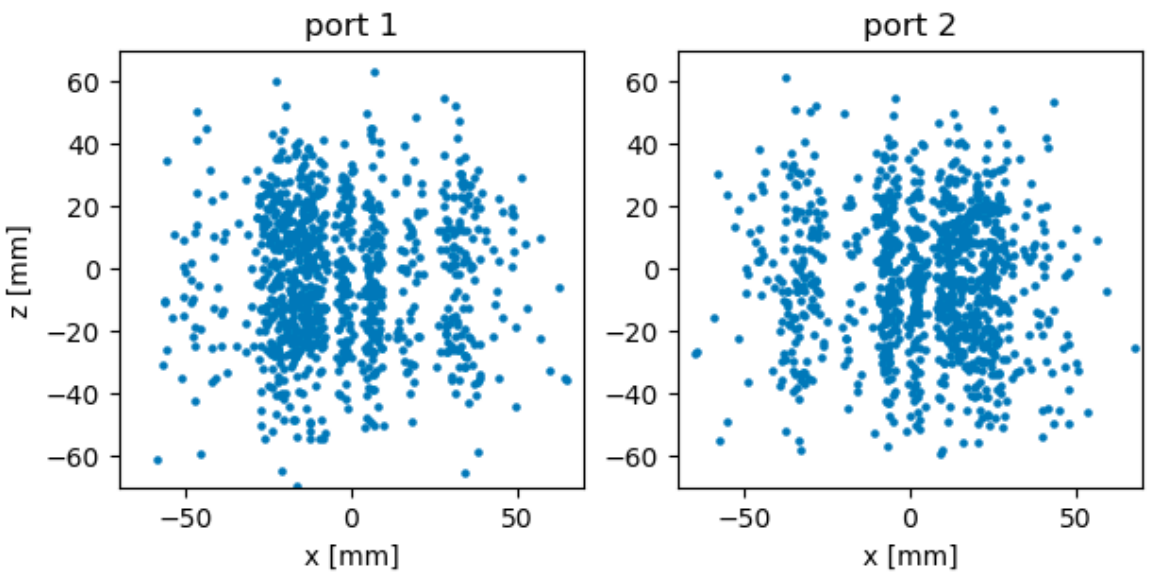
Let's talk about phase

Three contributions to accumulated phase difference in this framework:

- **Propagation phase:** integral of Lagrangian over trajectories
- **Separation phase:** spatial separation between endpoints
- **Laser phase:** non-uniformities in laser => phase differences from atom-laser interaction ("laser wavefront aberration")

All of these contribute to the fringes we observe when we measure the atom cloud.





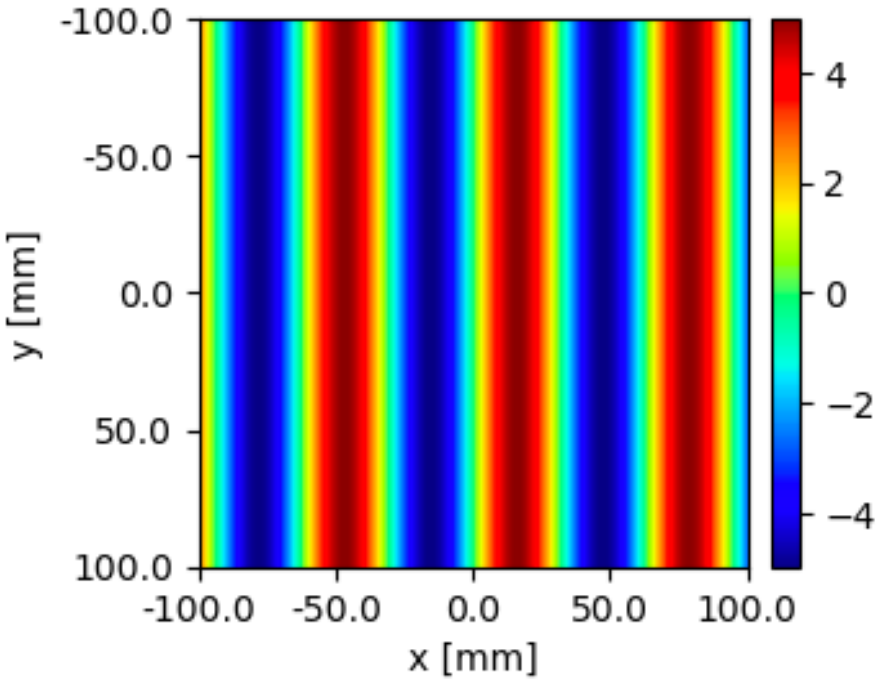
Fitting Laser Wavefront Aberration

The laser wavefront non-uniformity can be parametrized by some spatially dependent **phase function**

- This phase function has some form (Zernike polynomials, e.g.), and some corresponding set of parameters

Problem: given some real, measured data, we don't know what the parameters are

- Using our simulator and the measured data, can we extract (fit) the form of the laser wavefront aberration?



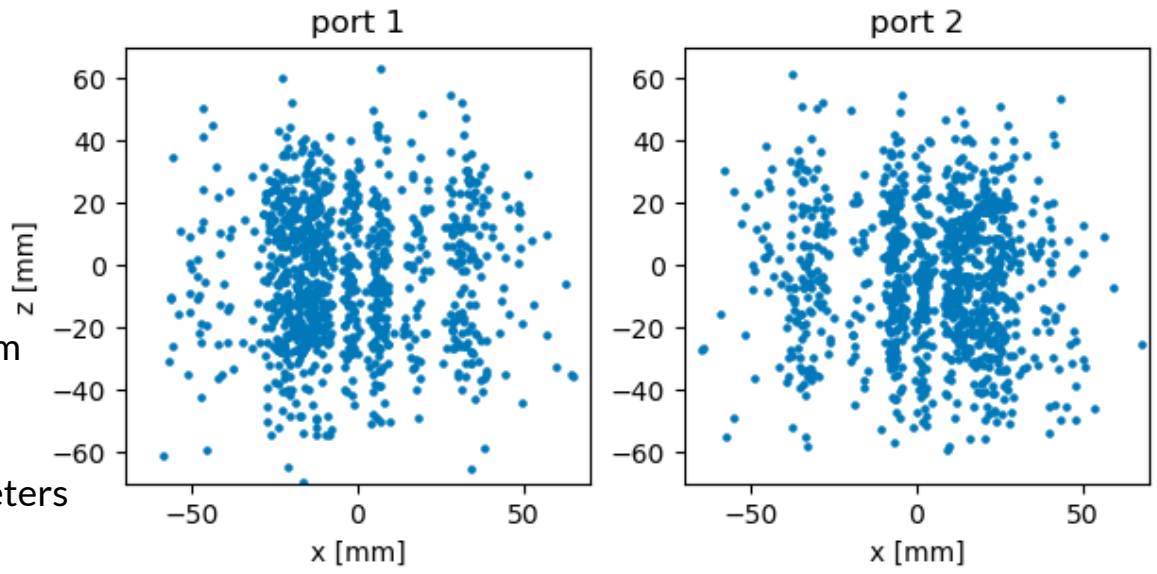
Aberration of form

$A \sin(k_x \cdot x)$

Unknown parameters

A: amplitude

k_x: spatial frequency



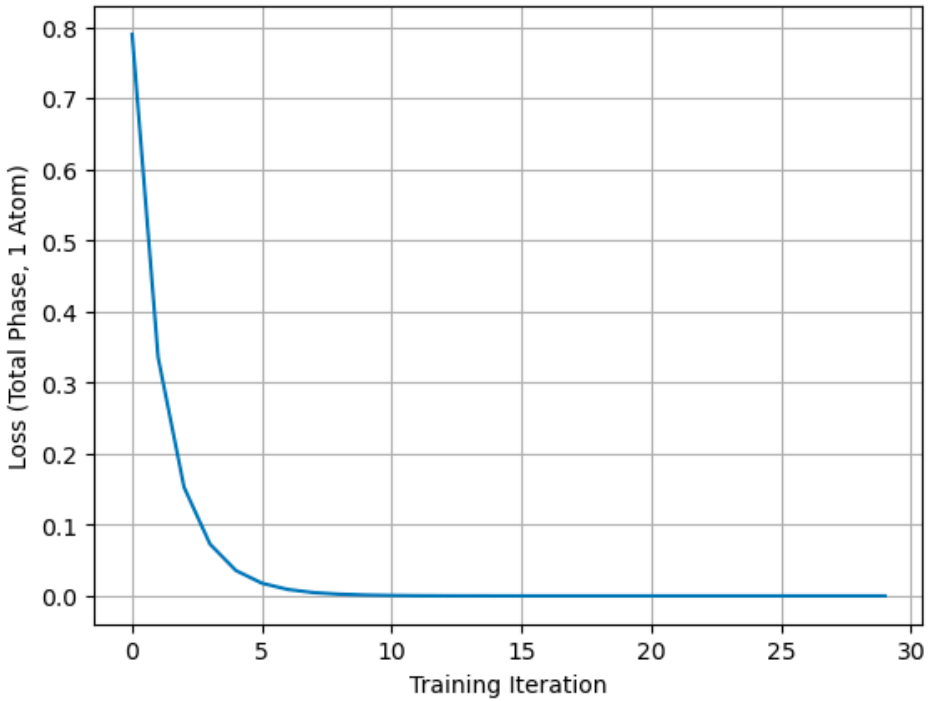
Step 1: Make the Simulator Differentiable

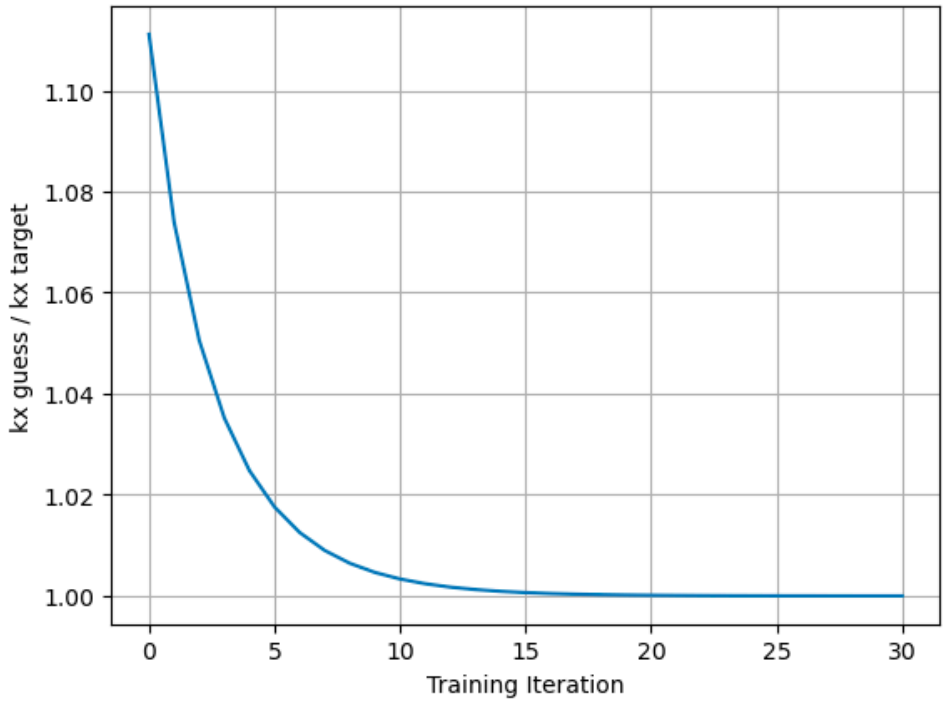
Automatic differentiation (see, e.g., previous talks from Michael) provides a way to get exact gradients of simulation **outputs** with respect to simulation **parameters**

- This allows for the automatic tuning of simulation parameters via **gradient descent** to meet some target objective (e.g. minimize a loss function)
- This sort of parameter tuning is a very efficient and powerful way to fit simulation to data!

Test:
Directly fit total phase of single atom.

Minimize $(\phi_{target} - \phi_{guess})^2$ by adjusting value of k_x in laser wavefront (via gradient descent)

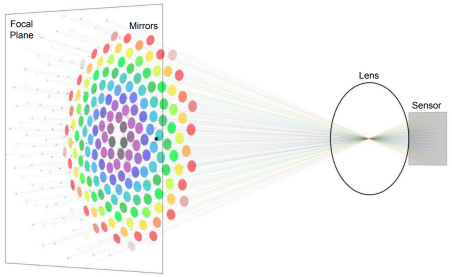


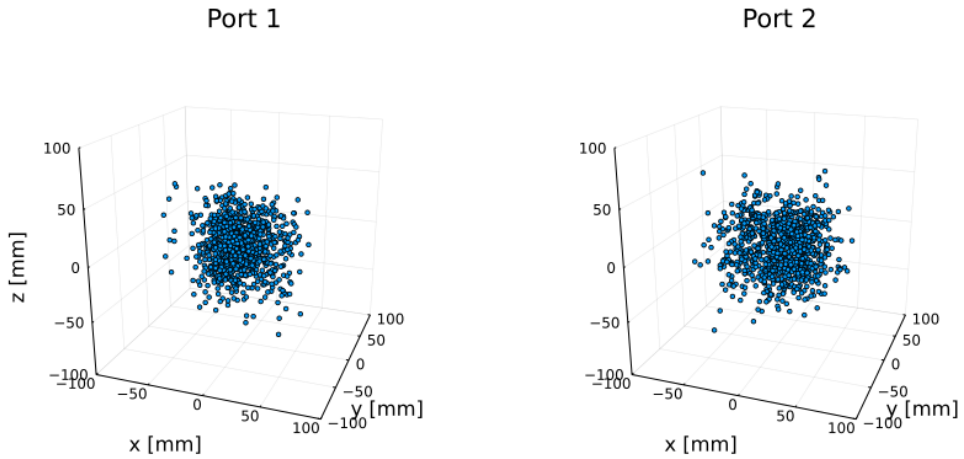


Step 2: How do we construct a loss function?

Gradient descent allows us to easily optimize, but what exactly should we optimize?

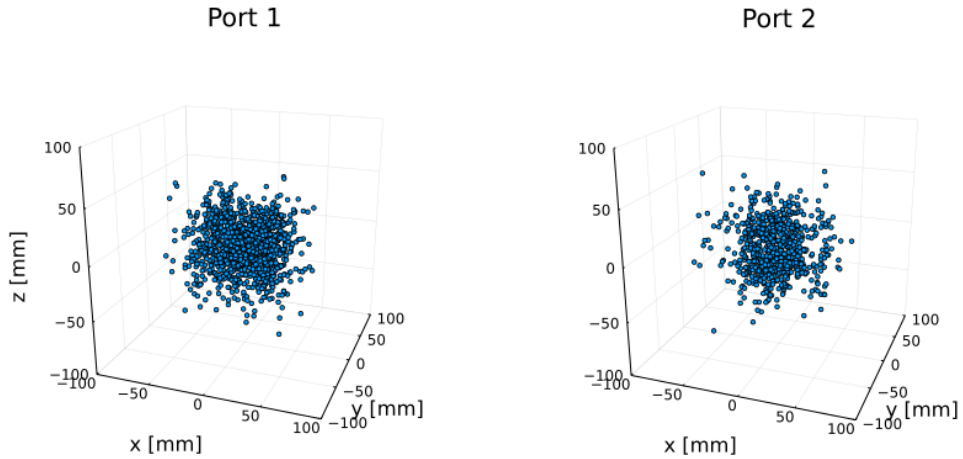
- Measurement is a set of 2D camera images
- Ongoing work: 2D images => 3D density (see, e.g. https://arxiv.org/abs/2205.11480)
  - Multi-view imaging may be crucial for extracting wavefront info!
- In the following, assume we have 3D information about final atom positions





$k_x = 100$

vs.



$k_x = 90$

?

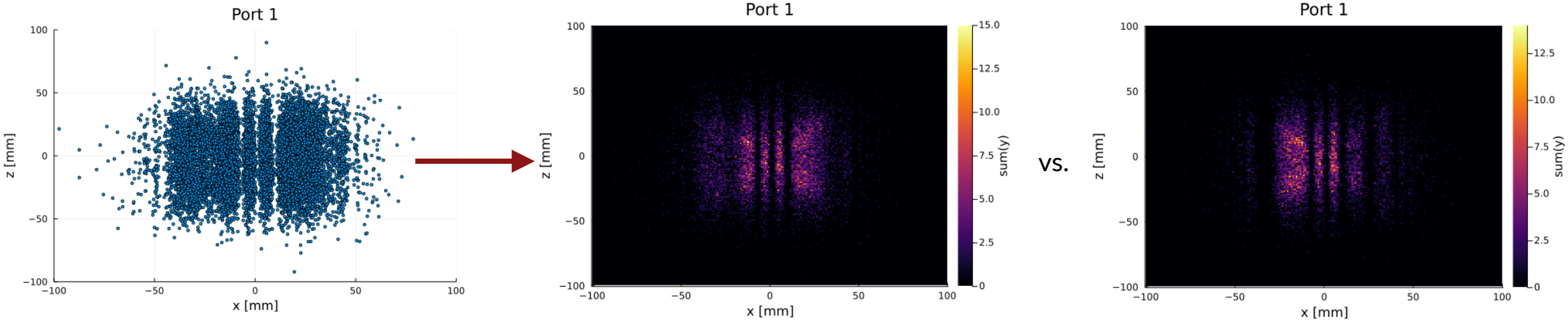
Loss Function Ideas: Direct Comparison

Procedure:

- Simulate a bunch of trajectories with some set of parameters
- Construct an empirical density from those trajectories (e.g. bin space into voxels)
- Compare density to measured and adjust parameters to match measurement

Drawbacks:

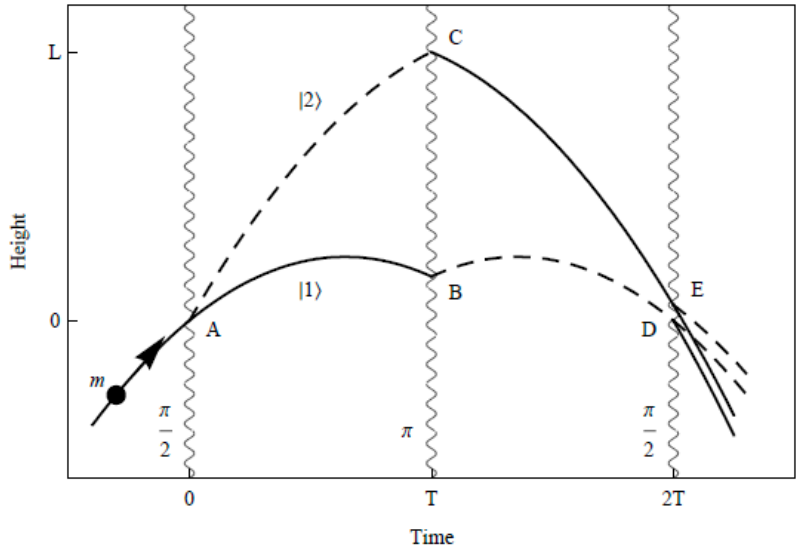
- Simulator is per-atom => construction of density impacts result (voxel size, statistics etc)
- Voxel binning breaks differentiability => need some tricks, approximations to nicely optimize

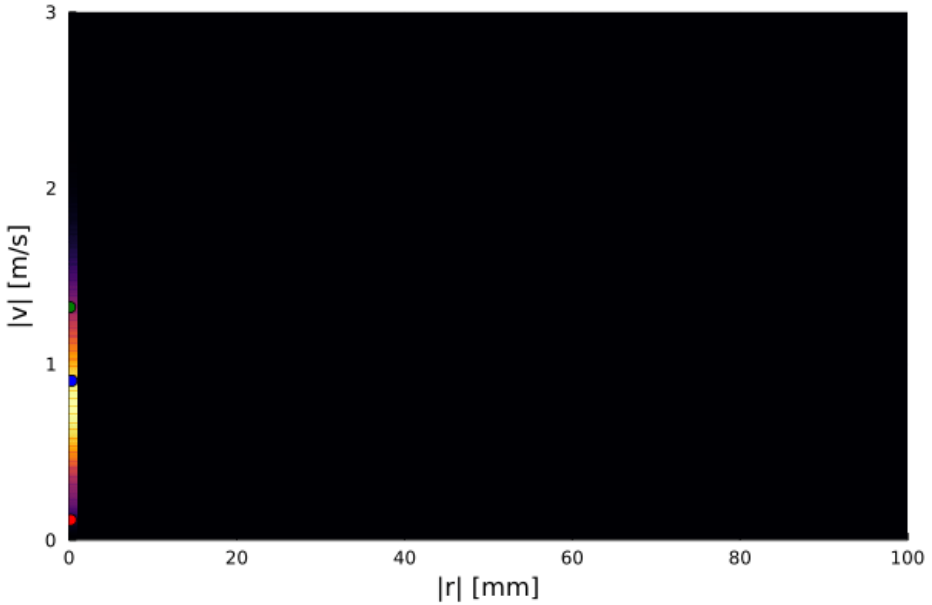


Towards an Atom Level Loss

Assume we know probability densities of initial position, $p(r_0)$ and velocity, $p(v_0)$.

- In our case: **classical equations of motion preserve probabilities**
  - $p(r_f, v_f) = p(r_0) \cdot p(v_0)$ for final position r_f, velocity v_f
- The quantum mechanical piece comes in with the phase probability
  - $p_k(r_f, v_f) = p(r_0) \cdot p(v_0) \cdot p_k(\phi(r_f, v_f))$, where k refers to port 1 or 2
- Summary: we can write down the probability of a measured 3D position if we know the corresponding initial position and the phase



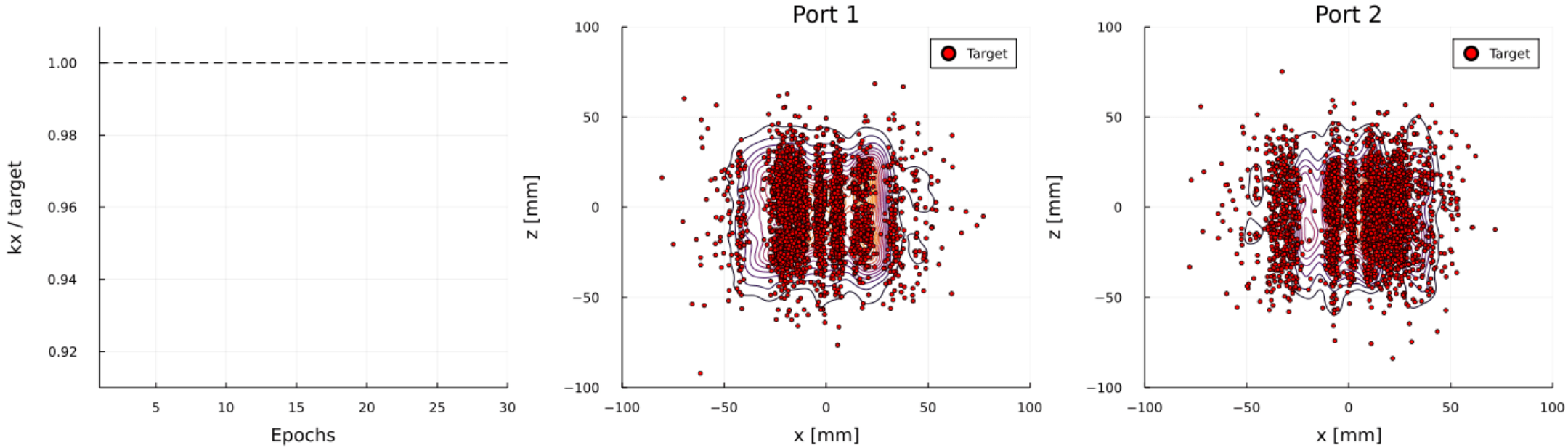


Timesteps A, C, E (before ports). Points transform along with distribution.

Maximum Likelihood Fitting

How do we use these probability properties? One way:

- From final position and velocity (r_f, v_f), can use a combination of reverse and forward solves of equations of motion to get r_0, v_0, ϕ (for some given simulation parameters)
- We can then construct a likelihood, and maximize this likelihood by adjusting parameters via gradient descent
- We can further marginalize over final velocities (e.g. with the help of neural networks), to do this procedure given only a measured set of final positions (similar to realistic case) — results below



Summary

We have:

- **Built a Julia simulator** for atoms in our atom interferometry system using a semi-classical approximation
- **Made this simulator differentiable**, allowing for automatic calculation of gradients of simulation outputs with respect to parameters
- **Demonstrated a maximum likelihood fit** of laser wavefront parameters with gradient descent using only measured final atomic positions

Next steps:

- Expanded fits: other methods, more complicated aberrations
- Improvements to simulator: optimized laser pulses
- Longer term: Incorporate measurement system/2D image → 3D density models